REMARKS

Reconsideration of the above-indicated patent application, as amended, is respectfully requested. The present amendment is responsive to the Non-Final Office Action mailed July 7, 2003. Claims 1-20 have been rejected. Accordingly, amended claims and supporting remarks are hereby presented that particularly point out and distinctly claim the subject matter that applicant regards as his invention. No new matter has been added.

THE INVENTION

The present invention includes embodiments directed to a method and apparatus for repeatedly executing an associated instruction. A count value, loaded into a register, indicates the number of times the associated instruction is to be repeatedly executed. A REPEAT instruction is then fetched and executed to indicate the associated instruction. The associated instruction is then fetched. The associated instruction is then repeatedly executed as many times as indicated by the count value. The present claims have been amended to explicitly recite these details such that the present claimed invention is very different from the prior art of record.

It should be appreciated that with the present method and apparatus, the associated instruction is fetched from memory only one time and then repeatedly executed until a desired count is reached. This is accomplished by placing the associated instruction in an "instruction register" in the CPU. The associated instruction is maintained in the instruction register for a suitable number of clock cycles, until the count value is reached. In this way, the present method and implementation perform repetitive instructions without "refetching" the instruction over and over from an instruction memory, as is done with typical "looping" and "unrolling" methods as

are known in the prior art (see, for example. page 7, lines 9 et seq. of the present disclosure). Instead, the processor program counter is effectively stalled on the same instruction which gets executed over and over again (see page 11, lines 22 and 23). Thus, the present system does not tie up the data bus, with instruction traffic, allowing other bus traffic each clock cycle, thereby improving efficiency. Also, since the instruction memory is accessed less frequently, power is conserved. The present system is also useful for context switching compared to prior art since the repeat state is concisely represented, only needing the count register and instruction register. When the state is saved and later restored, these two registers involve less overhead compared to maintaining state the state of the instruction buffer. The present embodiments are thus very different from the prior art relied on by the Examiner.

THE REJECTIONS UNDER 35 U.S.C. §102

Claims 1-7 had been rejected under Section 102(b) as allegedly being anticipated by Shridhar, et al. (U.S. Pat. No. 5,727,194). This rejection is respectfully traversed, particularly as applied to the amended claims as presented herewith.

Shridhar et al. is directed to a system including a repeat bit execution of repeat loops. It is quite clear from inspection that Shridhar et al. is a "pipelined" loop system in which a sequence of instructions is repeatedly fetched, decoded and executed. See, for example, col. 4, lines 44 et seq. and also col. 13, line 62 et seq. (relied on by the Examiner). It is apparent from Shridhar et al. that, upon detection of a "repeat" instruction, the first loop instruction is refetched (see col. 14, lines 1-5). Thus, Shridhar et al. can only be construed as simply a type of "looping" method such as is disclosed as prior art in the present specification (page 7, line 20) which involves re-fetching of instructions inside the loop to remove the branch loop overhead by unrolling. Shridhar et al.

thus cannot be relied on for anticipating a REPEAT instruction for fetching an associated instruction and repeatedly executing a single instruction without refetching, as is the subject of the present independent claims 1, 2 and 3, particularly as presently amended. Shridhar et al. also falls short with respect to context switching since this reference relies on additional registers, that is repeat start and repeat end address registers which are different than the present system (col. 2, line 46-50). Furthermore Shridhar et al. relies on augmentation of the instruction set with a repeat bit (col. 7, line 15) in order to implement. Thus Shridhar et al. requires a specialized assembler and recompilation (col. 6, lines 1-25), and in comparison, the present system simpler to implement. Thus, claims 1, 2, and 3 are believed to be allowable and dependent claims 4, 5, 6 and 7 are also believed to be allowable for at least the same reasons as the independent claims from which they depend. Therefore, reconsideration and withdrawal of this grounds of rejections is respectfully requested.

Claims 8-20 had been rejected under Section 102(b) as allegedly being anticipated by Kiuchi et al. (U.S. Pat. No. 5,579,493). This rejection is also respectfully traversed, particularly as applied to the amended claims as presented herewith.

Kiuchi et al. discloses a system that uses a loop buffer for storing instructions that are repeatedly executed, in order to decrease power consumption on the data processor. As shown in Fig. 1 and discussed inter alia at col. 8, lines 21 et seq., an instruction step to be repeatedly executed is read out of memory 101 and written into an instruction buffer 108, which holds a number of instructions to be looped. In the case of second and succeeding loops, the instructions are refetched from the instruction buffer 108 rather than the main instruction memory. In this way, Kiuch, et al. tries to reduce data bus traffic from the main memory 101.

However, it should be noted that Kiuchi et al. still relies on refetching in a "loop" type method. This is still very different from the present method and implementation in which a single instruction is fetched a single time and repeatedly executed from the instruction register until a desired count is reached. With the present system, the instruction register is effectively "frozen" during a repeat (not updated from any instruction memory) thereby forcing processor execution of the single repeated instruction for the desired number of repeat count cycles. Thus, Kiuchi et al.'s instruction buffer and related structures still fail to disclose a method and system that satisfies the requirements of the present independent claims. It should also be considered that Kiuchi et al. would have problems with context switching, since a change in context would require the Kiuchi et al. system to displace the instructions of the buffer, requiring new instructions to be fetched from main memory. In a context-switching environment, Kiuchi et al. must maintain the state of the instruction buffer (Fig. 1, item 108) and address registers (Fig. 2, 214 a, b). The "save" and "restore trashing" overhead can be severe in programming environments where context switches are common; especially when multiple programs are trying to share a common instruction buffer. Thus, the present system in beneficial in this respect as well.

In view of the above, Kiuchi et al. cannot be construed as satisfying the requirements of anticipation as applied against the present claims. It is therefore respectfully submitted that independent claims 8, 9, 10 and 15 distinguish patentably over Kiuchi et al. and it is believed that dependent claims 11-14 and 16-20 distinguish for at least the same reasons as the independent claims. Therefore, reconsideration and withdrawal of this grounds of rejection is respectfully requested.

In view of the foregoing it is respectfully submitted that the present claims, as currently amended, distinguish over the prior art. A notice to that effect is earnestly solicited. If the Examiner believes there are any further matters, which need to be discussed in order to expedite the prosecution of the present application, the Examiner is invited to contact the undersigned.

Respectfully submitted,
TUCKER ELLIS & WEST LLP

Jay P. Ryan
Agent for Applicant
(Registration No. 37,064)
1150 Huntington Building
925 Euclid Avenue
Cleveland, Ohio 44115-1475
Customer No. 23380
(216) 696-4396 (phone)
(216) 696-2645 (fax)